CLAIMS

1. A universal interface device, comprising:

a controller;

a configuration database coupled to the controller, said configuration database having stored therein a plurality of different configuration protocols for supporting a plurality of different peripheral devices;

a plurality of interconnection pads; a

a memory coupled to the interconnection pads and controller, the memory is programmable by the controller in order to support any of the different peripheral devices.

2. A universal interface device as defined in claim 1, wherein the controller comprises a state machine.

3. A universal interface device as defined in claim 1, further comprising a multiplexer coupled between the memory and the plurality of interconnection pads.

4. A universal interface device as defined in claim 1, further comprising a programmable clock coupled to the memory or the configuration database.

5. A universal interface device as defined in claim 1, wherein the controller selects a configuration protocol from amongst the plurality of configuration protocols in the configuration database, and uses the selected configuration protocol to configure the memory in order to support the peripheral device from amongst the plurality that is coupled to the plurality of interconnection pads.

6. A universal interface device as defined in claim 2, wherein the state machine includes a programmable routing and mapping scheme that allows the state machine to communicate with more than one peripheral device that is coupled to the plurality of interconnection pads.

7. A universal interface device as defined in claim 4, wherein the state machine can control the programmable clock so that it generates the necessary clock patterns required by the peripheral device coupled to the plurality of interconnection pads.

8. A universal interface as defined in claim 2, wherein the memory can be divided up by the state machine into two or more parts in order to support a peripheral device coupled to the interconnection pads that requires continuous transfer of data, the state machine switching between the two or more parts of the memory during data transfer to the peripheral device.

9. A universal interface as defined in claim 2, wherein the state machine sets a portion of the memory to provide a tri-state control if one or more of the plurality of interconnection pads have to function as both an input and an output.

10. A method for interfacing a controller with an electronic peripheral device that utilizes a predefined communications protocol, comprising the steps of:

selecting the appropriate configuration protocol for use with the electronic peripheral device from amongst a plurality of configuration protocols;

providing a memory; and

selectively interconnecting conductive pads coupled to the electronic peripheral device with sections of the memory programmed to support the configuration protocol used by the electronic peripheral device.

11. A method as defined in claim 10, further comprising the steps of:

receiving signals from the electronic peripheral device upon the conductive pads, and responsive thereto performing device-output-processing operations comprising:

directing the signals from the pads to the memory; and

storing the signals in the memory.

12. A method as defined in claim 10, further comprising the steps of:

receiving signals to be sent to the electronic peripheral device and responsive thereto

performing output signal processing operations comprising:

loading the output signals into the memory; and

outputting the output signals from memory with timing compliant with the identified communications protocol.

13. A method for universally interfacing a processor with an electronic peripheral device that utilizes a communications protocol of a prescribed list of multiple communications protocols, comprising operations of:

detecting that a peripheral device is coupled to the processor;

identifying the communication protocol used by the peripheral device coupled to the processor from a list of protocols supported by the processor;

receiving identification of first communications protocols utilized by a first peripheral device coupled to multiple conductive pads; and

selectively interconnecting the pads with memory input lines pursuant to the identified communications protocol.

14. A method as defined in claim 13, further comprising the steps of:

receiving output signals from the peripheral device upon the conductive pads, and responsive thereto performing device-output-processing operations comprising:

directing the output signals from the pads to the memory via the pads and memory input lines;

storing the output signals in the memory; and

directing the memory to output the stored signals to the processor.

15. A method as defined in claim 13, further comprising the steps of:

receiving processor output signals from the processor and responsive thereto performing processor-output-processing operations comprising:

loading the processor output signals into the memory; and

outputting the processor output signals from memory with timing compliant with the identified communications protocol.

16. The method of claim 13, where the peripheral device comprises multiple peripheral devices, and the operations further comprise time-dividing utilization of the pads between the multiple peripheral devices.

17. The method of claim 15, where:

the operations further include receiving identification of a second communications protocol different than the first communications protocol;

responsive to receiving the identification of the second communications protocol, conducting the processor-output-processing operations according to the second communications protocol.

18. A method of interfacing a processor with a peripheral device having one of multiple predetermined types, each type of peripheral device being designed to operate according to a different predefined communications protocol, comprising operations of:

providing an interface apparatus including a controller, memory, and multiple input/output nodes;

the controller receiving notification of presence of a peripheral device coupled to the input/output nodes, including the type of the peripheral device; and

responsive to the presence of the peripheral device, the controller operating the memory to simulate behavior of a dedicated interface between the processor and the peripheral device coupled to the input/output nodes.

19. A method as defined in claim 18, further comprising the step of:

recognizing by the controller any changes in peripheral device type coupled to the input/output nodes, and responsive to such changes the controller operating the memory to simulate behavior of a dedicated interface corresponding to the changed peripheral device.

20. A method of operating a universal interface which can support a plurality of communication protocols and including a multiplexer interposed between a memory and multiple input/output pads, the method comprising operations of:

identifying a communication protocol from amongst the plurality of communication protocols applicable to a peripheral device attached to the input/output pads;

retrieving pre-stored operating parameters corresponding to the identified communications protocol; and

configuring the multiplexer to selectively couple the input/output pads to the memory with mapping specified by the operating parameters.

21. A method as defined in claim 20, further comprising:

configuring a clock to provide a reference signal having a frequency specified by the operating parameters required by the identified communication protocol.

22. A method as defined in claim 20, further comprising the steps of:

loading data from a state machine into the memory; and

transmitting data from the memory to the pads via the multiplexer.

23. A method as defined in claim 20, further comprising the step of:

receiving data from the peripheral device into the memory via the multiplexer.